

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S): PARK, Jee-Hoon et al.
SERIAL NO.: Not Yet Assigned
FILED: Herewith
FOR: **APPARATUS AND METHOD FOR SYNCHRONIZING
PRESENCE ATTRIBUTE DATA BETWEEN TERMINAL
AND SERVER**
DATED: March 29, 2004


Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF PRIORITY DOCUMENTS

Sir:

Enclosed is a certified copy of Korean Patent Appln. No. 72645-
2003 filed on October 17, 2003, from which priority is claimed under 35 U.S.C.
§119.

Respectfully submitted,




Paul J. Farrell, Esq.
Reg. No. 33,494
Attorney for Applicant(s)

DILWORTH & BARRESE, LLP
333 Earle Ovington Blvd.
Uniondale, NY 11553
(516) 228-8484

CERTIFICATION UNDER 37 C.F.R. 1.10

I hereby certify that this New Application Transmittal and the documents referred to as enclosed therein are being deposited with the United States Postal Service in an envelope as "Express Mail Post Office to Addressee" Mail Label Number EL 995746055 US addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date listed below.

Dated: March 29, 2004



Michael J. Musella



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2003-0072645
Application Number

출원 년 월 일 : 2003년 10월 17일
Date of Application OCT 17, 2003

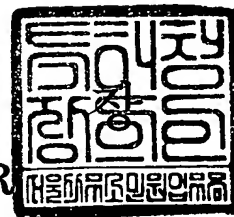
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 11 월 10 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】 특허출원서
【권리구분】 특허
【수신처】 특허청장
【참조번호】 0004
【제출일자】 2003.10.17
【국제특허분류】 H04M
【발명의 명칭】 단말장치와 서버간의 프레전스 정보 데이터 동기화를 위한 시스템 및 방법
【발명의 영문명칭】 SYSTEM AND METHOD FOR SYNCHRONIZING THE PRESENCE ATTRIBUTES BETWEEN THE TERMINAL AND DATA SERVER
【출원인】
【명칭】 삼성전자 주식회사
【출원인코드】 1-1998-104271-3
【대리인】
【성명】 이건주
【대리인코드】 9-1998-000339-8
【포괄위임등록번호】 2003-001449-1
【발명자】
【성명의 국문표기】 박지훈
【성명의 영문표기】 PARK, Jee Hoon
【주민등록번호】 740102-1109015
【우편번호】 151-755
【주소】 서울특별시 관악구 봉천3동 관악현대아파트 119-1111
【국적】 KR
【발명자】
【성명의 국문표기】 서호섭
【성명의 영문표기】 SE0, Ho Sub
【주민등록번호】 750910-1063711
【우편번호】 435-010
【주소】 경기도 군포시 당동 907번지 두산 아파트 103동 704호
【국적】 KR
【심사청구】 청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인
이건주 (인)

【수수료】

【기본출원료】	20 면	29,000 원
【가산출원료】	9 면	9,000 원
【우선권주장료】	0 건	0 원
【심사청구료】	12 항	493,000 원
【합계】	531,000 원	

【요약서】**【요약】**

본 발명은 단말장치와 서버간의 메신저 서비스를 위한 접속 시 프레전스 정보(Presence Attributes) 데이터 동기를 맞추기 위한 시스템 및 방법을 제공한다. 이를 위한 본 발명은 메신저 서비스를 위해 이전에 접속하였던 서버에 재접속하는 경우 업데이트 되어야 할 프레전스 정보 데이터를 상기 서버로 요청하여 상기 서버로부터 수신하는 업데이트 되어야 할 프레전스 정보 데이터에 의해 이전의 접속에 따라 저장되어 있는 프레전스 정보 데이터를 업데이트 하는 클라이언트 단말장치와, 상기 클라이언트 단말장치로부터 상기 업데이트 되어야 할 프레전스 정보 데이터 요구에 따라 현재 프레전스 정보 데이터 중 상기 이전의 접속 해제 이후부터 업데이트 된 프레전스 정보 데이터만을 상기 클라이언트 단말장치로 전송하는 서버로 구성됨을 특징으로 한다.

【대표도】

도 3

【색인어】

메신저, 프레전스 정보, 동기화.

【명세서】**【발명의 명칭】**

단말장치와 서버간의 프레전스 정보 데이터 동기화를 위한 시스템 및 방법{SYSTEM AND METHOD FOR SYNCHRONIZING THE PRESENCE ATTRIBUTES BETWEEN THE TERMINAL AND DATA SERVER}

【도면의 간단한 설명】

도 1은 일반적인 단말장치와 서버간의 통신을 위한 시스템을 도시하는 도면,

도 2는 종래의 단말장치와 서버간의 프레전스 정보 데이터 동기화 과정을 도시하는 신호 흐름도,

도 3은 본 발명의 실시 예에 따른 단말장치와 서버간의 프레전스 정보 데이터 동기화 과정을 도시하는 신호 흐름도,

도 4는 본 발명의 실시 예에 따라 단말장치에서 서버와 프레전스 정보 데이터를 동기화 하기 위한 제어 흐름도,

도 5는 본 발명의 실시 예에 따라 서버에서 단말장치와 프레전스 정보 데이터를 동기화 하기 위한 제어 흐름도.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <6> 본 발명은 단말장치와 서버간의 데이터 동기화를 위한 시스템 및 방법에 관한 것으로, 특히 단말장치와 서버간의 메신저 서비스를 위한 접속 시 프레전스 정보(Presence Attributes) 데이터 동기를 맞추기 위한 시스템 및 방법에 관한 것이다.
- <7> 일반적으로, 클라이언트 단말장치와 서버의 데이터 호환성을 위하여 이들 시스템 간에 접속이 있는 경우에는 서로간의 데이터 동기화가 필요하다. 이러한 클라이언트 단말장치와 서버간의 접속을 위한 시스템을 도시하고 있는 도 1을 참조하면, 클라이언트 단말장치(100, 102, 104)는 메신저 서비스를 제공할 수 있는 단말장치로 인터넷 망(108) 또는 이동통신 무선망(106)을 통해 서버(110)와 접속할 수 있다. 또한, 메신저 서비스를 제공하기 위한 서버(110)는 프레전스 정보 데이터 베이스(112)를 구비하여 해당 클라이언트 단말장치(10)의 친구목록정보, 목록에 존재하는 각각의 친구들의 상태정보, 친구들의 주소정보, 컨택(contact) 정보 등과 같은 프레전스 정보를 저장하고 있다. 즉, 여기서 프레전스 정보란 메신저 서비스를 수행하기 위해 클라이언트 단말장치(10)와 서버(110)가 저장하고 있는 친구목록정보, 목록에 존재하는 각각의 친구들의 상태정보, 친구들의 주소정보, 컨택(contact) 정보 등의 정보 데이터이다. 클라이언트 단말장치(10)가 서버(110)로 접속하고, 메신저 서비스를 받기 위해 서버(110)로 프레전스 정보를 요청하고, 서버(110)로부터 프레전스 정보를 수신하게 된다.

- <8> 그러면, 종래의 클라이언트 단말장치(10)와 서버(110)간의 프레전스 정보 데이터 동기화 과정에 대하여 도 2를 참조하여 설명하도록 한다. 도 2는 종래의 클라이언트 단말장치와 서버간의 프레전스 정보 동기화 과정을 도시하는 신호 흐름도이다.
- <9> 먼저, 클라이언트 단말장치(10)가 사용자로부터 메신저 서비스를 위한 입력이 있는 경우 서버(110)로 204단계에서 서버 접속 요청(Login request)신호를 전송한다. 이를 수신한 서버(110)는 접속을 요청한 해당 클라이언트 단말장치(10)를 확인한 후 접속 요청에 대한 응답신호를 전송한 후, 클라이언트 단말장치(10)와 서버(110)간에 사용자 서비스정보, 클라이언트 단말장치(10)정보를 송수신한다. 이후 206단계에서 클라이언트 단말장치(10)가 서버(110)로 프레전스 정보를 요청(Presence Attribute values request)하고, 이를 수신한 서버(110)는 해당 클라이언트 단말장치(10)에 대한 프레전스 정보를 프레전스 정보 데이터 베이스(112)로부터 독출하여 클라이언트 단말장치(10)로 전송한다.
- <10> 예를 들어, 3시에 서버(110), 클라이언트 단말장치(10) 사이의 통신이 정상적, 혹은 비정상적으로 종료되었을 경우 이후 4시에 다시 클라이언트 단말장치(10)가 서버(110)로 재접속하게 되면 한 시간 사이에 클라이언트 단말장치(10)가 가지고 있던 프레전스 데이터가 더 이상 유효한지 알 수 없기 때문에 모든 데이터를 서버(110)로부터 확인하여야 한다. 이 때문에 비록 1시간 동안 변동이 없는 데이터도 서버(110)로부터 다시 확인해야 하는 절차를 거쳐야 하는 문제점이 있었다. 즉, 상기한 예와 같이 3시에 클라이언트 단말장치(10)가 서버(110)와 접속 해제되고, 4시에 다시 재접속하는 경우 서버(110)에 저장된 해당 클라이언트 단말장치(10)의 프레전스 정보가 3시와 4시가 동일할 경우에도 클라이언트 단말장치(10)는 모든 프레전스 정보를 요청하고 서버(110)는 3시와 동일한 프레전스 정보를 클라이언트 단말장치(10)로 모두 전송하여야만 하는 경우가 발생하였다. 또한, 3시와 4시에 해당 클라이언트 단말장치(10)에 대한

프레전스 정보 중 업데이트 된 정보가 적을 경우에도 프레전스 정보를 모두 전송하여야 하기 때문에 불필요한 데이터 전송으로 인해 트래픽이 증가하고, 많은 시간이 소요되었다.

【발명이 이루고자 하는 기술적 과제】

- <11> 상술한 바와 같이 종래에는 클라이언트 단말장치(10)가 서버(110)에 접속할 때마다 클라이언트 단말장치(10)는 자신의 모든 프레전스 정보를 요청하고, 이를 수신하여 정보를 갱신한다. 예를 들어 클라이언트 단말장치(10)가 서버(110)와 접속 해제된 시점에서의 프레전스 정보와 다시 접속하는 시점에서의 프레전스 정보가 동일하거나 업데이트 된 정보가 적을 경우에도 서버(110)는 프레전스 정보를 클라이언트 단말장치(10)로 모두 전송하였기 때문에 불필요한 트래픽 및 시간이 소요되었다.
- <12> 따라서 본 발명의 목적은 클라이언트 단말장치가 서버로 재접속시 서버가 클라이언트 단말장치로 업데이트 해야 할 프레전스 정보 데이터만을 전송하여 보다 효율적인 데이터 동기화를 수행할 수 있는 시스템 및 방법을 제공함에 있다.
- <13> 본 발명의 다른 목적은 클라이언트 단말장치가 서버로 재접속시 서버는 업데이트 해야 할 프레전스 정보 데이터만을 클라이언트 단말장치로 송신하여 불필요한 데이터 전송으로 인한 트래픽 증가를 방지할 수 있는 시스템 및 방법을 제공함에 있다.
- <14> 본 발명의 또 다른 목적은 클라이언트 단말장치가 서버로 재접속시 클라이언트 단말장치는 서버로부터 업데이트 해야 할 프레전스 정보 데이터만을 수신하여 빠른 시간안에 현재 상태 프레전스 정보 데이터를 유효한 데이터로 바꿀 수 있는 시스템 및 방법을 제공함에 있다.

【발명의 구성 및 작용】

- <15> 상기한 목적들을 달성하기 위한 본원발명은 메신저 서비스를 위해 이전에 접속하였던 서버에 재접속하는 경우 업데이트 되어야 할 프레전스 정보 데이터를 상기 서버로 요청하여 상기 서버로부터 수신하는 업데이트 되어야 할 프레전스 정보 데이터에 의해 이전의 접속에 따라 저장되어 있는 프레전스 정보 데이터를 업데이트 하는 클라이언트 단말장치와, 상기 클라이언트 단말장치로부터 상기 업데이트 되어야 할 프레전스 정보 데이터 요구에 따라 현재 프레전스 정보 데이터 중 상기 이전의 접속 해제 이후부터 업데이트 된 프레전스 정보 데이터만을 상기 클라이언트 단말장치로 전송하는 서버로 이루어짐을 구비함을 특징으로 한다.
- <16> 상기한 목적들을 달성하기 위한 본원발명은 클라이언트 단말장치와 상기 클라이언트 단말장치로 메신저 서비스를 제공하는 서버로 구성되는 시스템에서 상기 클라이언트 단말장치와 상기 클라이언트 단말장치간에 프레전스 정보 데이터 동기화를 위한 방법에 있어서,
- <17> 상기 메신저 서비스를 위해 이전에 접속하였던 서버에 재접속하는 경우 업데이트 되어야 할 프레전스 정보 데이터를 상기 서버로 요청하는 과정과,
- <18> 상기 서버는 상기 클라이언트 단말장치로부터 상기 업데이트 되어야 할 프레전스 정보 데이터 요구에 따라 현재 프레전스 정보 데이터 중 상기 이전의 접속 해제 이후부터 업데이트 된 프레전스 정보 데이터만을 상기 클라이언트 단말장치로 전송하는 과정과,
- <19> 상기 클라이언트 단말장치가 상기 서버로부터 수신하는 업데이트 되어야 할 프레전스 정보 데이터에 의해 이전의 접속에 따라 저장되어 있는 프레전스 정보 데이터를 업데이트 하는 과정으로 이루어짐을 구비함을 특징으로 한다.

- <20> 이하, 본 발명에 따른 바람직한 실시 예를 첨부한 도면의 참조와 함께 상세히 설명한다. 본 발명을 설명함에 있어서, 관련된 공지기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단된 경우 그 상세한 설명은 생략한다.
- <21> 클라이언트 단말장치(10)와 서버(110)간이 통신이 해제되는 순간에 서버(110)와 클라이언트 단말장치(10)는 이미 동일한 데이터를 저장하고 있다. 본 발명에서는 이와 같이 서버(110)와 클라이언트 단말장치(10)가 해제되는 순간에 저장하고 있는 동일 데이터를 이용하여 이 데이터를 구분할 수 있는 동일한 키 값을 이용하여 해제 후 재접속이 있는 경우에 클라이언트 단말장치(10)는 서버(110)에 키 값을 전송하고, 서버(110)는 그 키 값을 수신하여 서버에 동일한 키 값과 그에 해당하는 데이터를 가지고 있다면, 클라이언트 단말장치(10)에 변경되어야 하는 데이터만 전송한다.
- <22> 먼저, 클라이언트 단말장치(10)와 서버(110)간의 통신을 위한 시스템을 도시하는 도 1을 참조하여 설명한다.
- <23> 클라이언트 단말장치(10)는 컴퓨터(100), 휴대폰(102), PDA(Personal Digital Assistant)(104)등과 같이 메신저 서비스를 제공할 수 있는 모든 단말기를 포함한다. 이러한 클라이언트 단말장치(10)는 친구리스트 목록과 그 상태 등과 같은 프레전스 정보 데이터를 서버(110)와 접속이 종료되었을 때 가장 마지막으로 서버(110)와 통신이 성공했을 때의 상태로 저장한다. 또한, 서버(110)와 클라이언트 단말장치(10)의 이전 연결(session)의 고유한 아이디인 세션 아이디(session id)와, 친구리스트의 목록과 상태를 저장하고 있는 클라이언트 단말장치(10)의 고유 아이디(client id)와, 클라이언트 단말장치(10)와 서버(110)간의 통신 트랜잭션 아이디(transaction id)를 저장한다. 여기서 트랜잭션이란 클라이언트 단말장치(10)의 요청에 대한 서버(110)의 응답, 혹은 서버(110)의 요청에 대한 클라이언트의 응답(10)을 말한다. 하나

의 서버(110)와 클라이언트 단말장치(10)의 연결(session)될 때마다 고유한 트랜잭션 아이디(transaction id)가 부여된다. 이때, 클라이언트 단말장치(10)는 서버(110)와 클라이언트 단말장치(10)사이의 통신 중 서버(110)와 클라이언트 단말장치(10)사이의 요청과 응답이 모두 완료된 경우인 정상적으로 종료된 트랜잭션의 아이디를 저장한다.

<24> 또한, 클라이언트 단말장치(10)는 재접속 시 상기와 같이 저장된 세션 아이디(session id)와, 클라이언트 단말장치(10)의 고유 아이디(client id), 서버(110)와의 최종으로 통신한 트랜잭션 아이디(transaction id)를 포함하는 키를 생성하여 서버(110)로 전송한다.

<25> 그러면, 이제 서버(110)를 살펴보면, 클라이언트 단말장치(10)와 인터넷망으로 연결되어 클라이언트 단말장치(10)로부터 메신저 서비스 요청이 있는 경우 메신저 서비스를 제공하는 서버이다.

<26> 우선, 서버(110)는 프레전스 정보 데이터 베이스(112)에 각 클라이언트 단말장치(10)로 메신저 서비스를 제공하기 위한 정보 데이터를 저장한다. 서버(110)는 프레전스 정보 데이터 베이스(112)에 각 사용자에게 따른 사용자 아이디(user id) 및 아이디에 따른 비밀번호, 클라이언트 단말장치(10)의 아이디(client id), 클라이언트와의 접속을 나타내는 세션 아이디(session id)에 따라 정의되는 친구리스트의 목록과 친구의 상태 즉, 프레전스 정보를 클라이언트 단말장치(10)와 접속이 종료되었을 때 가장 마지막으로 클라이언트 단말장치(10)와 통신이 성공했을 때의 상태로 저장한다. 또한, 서버(110)는 프레전스 정보 데이터 베이스(112)에 클라이언트 단말장치(10)와의 접속 전 최종 연결(session)의 고유한 아이디인 세션 아이디(session id)와 친구리스트의 목록과 상태를 저장하고 있는 클라이언트 단말장치(10)의 고유 아이디(client id), 클라이언트 단말장치(110)와의 통신 중 정상적으로 종료된 트랜잭션의 최종 트랜잭션 아이디(transaction id)를 저장한다. 또한, 서버(110)는 클라이언트 단말장치(10)

로부터 재접속 요구를 수신하였을 경우 수신한 키 값에 포함되어 있는 세션 아이디(session id)와, 클라이언트 단말장치(10)의 고유 아이디(client id), 서버(110)와의 최종으로 통신한 트랜잭션 아이디(transaction id)와 프레전스 정보 데이터 베이스에 저장되어 있는 키 값과 동일한지를 비교한다.

<27> 만약 클라이언트 단말장치(10)로부터 재접속시 수신한 키 값과 동일한 키 값이 프레전스 정보 데이터 베이스(112)에 저장되어 있으면, 서버(110)는 사용자의 친구리스트에 포함된 친구들의 현재 상태인 프레전스 정보 데이터(presence attributes values at this moment)와 동일한 키 값에 대응되어 저장된 접속 해제 시 가장 마지막으로 유효한 프레전스 정보 데이터(presence attributes values linked with the synchronization key)를 비교하여 차이가 있는 영역(field)을 독출한다. 그리고 차이가 있는 영역을 프레전스 정보 요청에 대한 응답 데이터 영역에 포함시켜 단말장치(10)로 전송한다. 이때, 현재 상태의 프레전스 정보 데이터와 동일한 키 값에 대응되어 저장되어 있는 프레전스 정보 데이터 간에 업데이트(up data) 된 정보가 없는 경우에는 응답 데이터 영역에 널(null) 상태로 전송한다. 이 후 서버(10)와 클라이언트 단말장치(110)간의 프레전스 정보 데이터를 동기화한 후, 동기화에 사용된 키 값과 그에 대응되는 프레전스 데이터를 제거하여도 된다.

<28> 또한, 클라이언트 단말장치(10)로부터 재접속 수신한 키 값과 동일한 키 값이 프레전스 정보 데이터 베이스(112)에 저장되어 있지 않으면, 서버(110)는 프레전스 정보 동기화 실패하였음을 알리는 메시지(status in failure case)를 클라이언트 단말장치(10)에 전송한다. 또한, 이를 수신한 클라이언트 단말장치(10)는 저장하고 있던 키와 그에 대응되는 데이터가 더 이상 유효하지 않으므로 삭제한다. 그리고 일반적인 프레전스 동기화 과정으로 서버(110)로 프레전스 정보를 요청하고, 서버(110)로부터 사용자의 친구리스트에 포함된 친구들의 모든 현재 상

태 데이터(presence attributes values)를 수신한 후 프레전스 데이터 업데이트 하여 서버(110)와 동기화한다.

<29> 그러면, 이제 상기한 클라이언트 단말장치(10)와 서버(110)간에 재접속 시 프레전스 정보 데이터 동기화과정에 대하여 도 3을 참조하여 설명한다.

<30> 우선, 클라이언트 단말장치(10)와 서버(110)가 연결되면 클라이언트 단말장치(10)와 서버(110)는 연결되었을 때 연결에 대한 세션 아이디(session id)와 클라이언트 아이디(client id)를 공유한다. 또한, 이 경우 클라이언트 단말장치(10)와 서버(110)는 접속 해제 시 가장 마지막으로 상호간의 성공적인 트랜잭션 아이디(transaction id)를 각각 저장하고 있다. 여기서, 클라이언트 단말장치(10)와 서버(110)가 저장하고 있는 트랜잭션 아이디(transaction id)는 정상적인 접속 해제 시에는 로그아웃요청 혹은 연결종료요청이 가장 마지막의 성공적인 트랜잭션 이고, 비정상적인 접속 해제 시에는 연결이 실패한 직전의 트랜잭션이 가장 마지막의 성공적인 트랜잭션이다. 본 발명에서는 클라이언트 단말장치(10)와 서버(110) 각각이 사용자 친구 리스트 및 그의 상태 등의 프레전스 정보 데이터를 공유하고 있다가 접속 해제가 된 경우임을 가정한다. 이와 같이 접속 해제될 시 클라이언트 단말장치(10)와 서버(110)는 각각 저장하고 있는 세션 아이디(session id), 클라이언트 아이디(client id), 트랜잭션 아이디(transaction id)로 구성되는 프레전스 정보 데이터 동기화를 위한 동기 키값을 생성하여 저장한다. 또한, 클라이언트 단말장치(10)와 서버(110)는 상기와 같이 생성된 동기 키와 그에 대응되는 트랜잭션 아이디(transaction id)에 해당하는 사용자의 친구 리스트의 상태 데이터인 프레전스 정보 데이터를 각각 저장한다.

<31> 이후, 클라이언트 단말장치(10)와 서버(110)간의 접속 해제 후, 300단계에서 클라이언트 단말장치(10)가 서버(110)에 재접속을 요청하면, 우선 이를 수신한 서버(110)는 접속을 요청

한 해당 단말장치(10)를 확인한 후 접속 요청에 대한 응답신호를 전송한 후, 클라이언트 단말장치(10)와 서버(110)간에 사용자 서비스정보, 클라이언트 단말장치(10)정보 등을 송수신한다. 그런 후, 304단계에서 클라이언트 단말장치(10)는 서버(110)가 상기와 같이 생성된 동기 키(synckey)를 이용하여 프레전스 정보를 동기화하는 서비스를 제공하는 지를 확인하는 메시지를 서버(110)로 전송한다. 이를 수신한 서버(110)는 동기 키(synckey)를 이용하여 프레전스 정보를 동기화하는 서비스를 제공하는 경우에는 응답 메시지를 306단계에서 전송한다.

<32> 306단계에서 클라이언트 단말장치(10)가 프레전스 정보를 동기화하는 서비스를 제공 확인에 대한 응답 메시지를 수신하면, 클라이언트 단말장치(10)는 사용자의 친구 리스트 및 그의 상태 정보인 프레전스 정보를 서버(110)로 요청한다. 이때 본 발명에서는 세션 아이디(session id), 클라이언트 아이디(client id), 트랜??션 아이디(transaction id)로 구성되는 프레전스 정보 데이터 동기화를 위해 상기와 같이 생성된 동기 키(synckey)를 전송함으로써 프레전스 정보를 요청하는 것이다.

<33> 그러면, 서버(110)는 클라이언트 단말장치(10)로부터 수신한 동기 키와 동일한 동기 키가 프레전스 정보 데이터 베이스(112)에 저장되어 있는지를 검사하여 동일한 동기 키가 존재할 경우에는 314단계로 진행하고, 그렇지 않으면 312단계로 진행하여 프레전스 정보 동기화 실패 메시지를 클라이언트 단말장치(10)로 전송한다. 또한, 이를 수신한 클라이언트 단말장치(10)는 저장하고 있던 키와 그에 대응되는 데이터가 더 이상 유효하지 않으므로 삭제한다. 그리고 일반적인 프레전스 동기화 과정으로 서버(110)로 프레전스 정보를 요청하고, 서버(110)로부터 사용자의 친구리스트에 포함된 친구들의 모든 현재 상태 데이터(presence attributes values)를 수신한 후 프레전스 데이터 업데이트하여 서버(110)와 동기화한다.

<34> 한편, 클라이언트 단말장치(10)로부터 수신한 키 값과 동일한 키 값이 프레전스 정보 데이터 베이스(112)에 저장되어 있으면, 314단계로 진행하여 서버(110)는 사용자의 친구리스트에 포함된 친구들의 현재 상태인 프레전스 정보 데이터(presence attributes values at this moment)와 동일한 키 값에 대응되어 저장된 접속 해제시 가장 마지막으로 유효한 프레전스 정보 데이터(presence attributes values linked with the synchronization key)를 비교하여 갱신된 영역(field)을 독출한다. 이후 316단계에서 갱신된 영역을 프레전스 정보 요청에 대한 응답 데이터 영역에 포함시켜 단말장치(10)로 전송한다. 이때, 현재 상태의 프레전스 정보 데이터와 동일한 키 값에 대응되어 저장되어 있는 프레전스 정보 데이터 간에 업데이트(up data)된 정보가 없는 경우에는 응답 데이터 영역에 널(null) 상태로 전송한다. 이 후 서버(110)와 클라이언트 단말장치(110)간의 프레전스 정보 데이터를 동기화한 후, 동기화에 사용된 키 값과 그에 대응되는 프레전스 데이터를 제거하여도 된다.

<35> 상기한 도 4와 같이 클라이언트 단말장치(10)와 서버(110)간의 프레전스 정보 동기화를 위해 동기 키 값을 이용하여 재접속 시 접속 해제된 시점에 저장되어 있는 프레전스 정보에서 현재 업데이트 된 프레전스 정보만을 서버(110)가 클라이언트 단말장치(10)로 전송할 수 있다.

<36> 그러면, 이제 클라이언트 단말장치(10)가 상기한 바와 같이 동기 키 값을 이용하여 서버(110)와의 프레전스 정보 동기를 맞추는 과정에 대하여 도 4를 참조하여 설명하도록 한다. 도 4는 본 발명의 실시 예에 따라 단말장치(10)에서 서버(110)와 동기를 맞추기 위한 제어 흐름도이다.

<37> 먼저, 400단계의 대기상태에서 402단계로 진행하면, 사용자에게 의해 메신저 서비스를 받기 위해 서버(110)로의 접속을 위한 입력이 있는 경우 404단계로 진행하여 서버(110)로 접속을 요청한다. 이후 406단계에서 서버(110)로부터 접속 요청에 대한 응답이 있는 경우에는 408단

계로 진행하여 접속을 위한 사용자 서비스정보, 클라이언트 단말장치(10) 정보를 서버(110)로 전송하고, 410단계로 진행하여 서버(110)로 동기 키 값을 이용하여 프레전스 정보 동기화 서비스 기능을 수행 가능한지를 확인 요청하고, 412단계에서 서버(110)로부터 동기 키 값을 이용하여 프레전스 정보 동기화 서비스 기능을 수행 가능함에 대한 응답 메시지가 수신되었으면 414단계로 진행하고, 그렇지 않으면 422단계로 진행하여 일반적인 프레전스 동기화 과정을 수행한다. 일반적인 프레전스 동기화 과정이란 상기에서 설명한 바와 같이 클라이언트 단말장치(10)가 서버(110)로 프레전스 정보를 요청하고, 서버(110)로부터 사용자의 친구리스트에 포함된 친구들의 모든 현재 상태 데이터(presence attributes values)를 수신한 후 프레전스 데이터 업데이트하여 서버(110)와 동기화하는 과정이다.

<38> 한편, 412단계에서 414단계로 진행하면, 세션 아이디(session id), 클라이언트 아이디(client id), 트랜잭션 아이디(transaction id)로 구성되는 동기 키(synckey)를 생성하고, 416단계로 진행하여 생성된 동기 키(synckey)를 프레전스 정보 동기화 요청 메시지에 포함시켜 서버(110)로 전송한다. 이후, 418단계에서 서버(110)로부터 갱신할 프레전스 정보 데이터가 수신되면, 420단계로 진행하여 저장된 프레전스 정보에서 수신한 프레전스 정보 부분만을 갱신하여 저장한다.

<39> 클라이언트 단말장치(10)가 상기한 바와 같이 동기 키 값을 이용하여 서버(110)로 프레전스 정보를 요청하는 경우 서버(110)에서 클라이언트 단말장치(10)로 프레전스 정보를 전송하여 상호간의 프레전스 동기를 맞추는 과정에 대하여 도 5를 참조하여 설명하도록 한다. 5는 본 발명의 실시 예에 따라 서버(110)에서 클라이언트 단말장치(10)와 동기를 맞추기 위한 제어 흐름도이다.

<40> 먼저, 500단계에서 대기상태를 유지하는 중 502단계에서 클라이언트 단말장치(10)로부터 접속(login) 요청이 수신되는지를 검사한다. 접속(login) 요청이 수신되었으면 504단계로 진행하여 접속(login) 요청에 대한 응답을 전송하고, 506단계에서 클라이언트 단말장치(10)로부터 동기 키 값을 이용한 프레전스 동기화 서비스가 가능한지를 확인 요청이 있는 경우에는 508단계로 진행하고, 그렇지 않으면 522단계로 진행한다. 508단계로 진행하면, 서버(110)는 동기 키 값을 이용한 프레전스 동기화 서비스가 가능한지를 검사하여 가능한 경우에는 510단계로 진행하고, 그렇지 않으면 520단계로 진행하여 동기 키 값을 이용한 프레전스 동기화 서비스가 불가능함을 알리는 메시지를 클라이언트 단말장치(10)로 전송하고 종료한다. 또한, 508단계에서 동기 키 값을 이용한 프레전스 동기화 서비스가 가능한 경우에는 510단계로 진행하여 동기 키 값을 이용한 프레전스 동기화 서비스가 가능한지를 확인 요청에 대한 응답 메시지를 전송한다. 이후 512단계로 진행하여 클라이언트 단말장치(10)로부터 동기 키 값이 포함된 프레전스 정보 요청이 있는 경우 514단계로 진행하여 수신한 동기 키 값과 일치하는 동기 키 값이 프레전스 정보 데이터 베이스(112)에 저장되어 있는지를 검사하여 동일한 동기 키 값이 있는 경우에는 516단계로 진행하여 동기 키 값에 따른 최종 프레전스 정보와 현재 프레전스 정보를 비교하여 갱신된 프레전스 정보를 독출한다. 이후 518단계로 진행하여 갱신해야하는 프레전스 정보 데이터를 클라이언트 단말장치(10)로 전송한다. 즉, 사용자의 친구리스트에 포함된 친구들의 현재 상태인 프레전스 정보 데이터(presence attributes values at this moment)와 동일한 키 값에 대응되어 저장된 접속 해제시 가장 마지막으로 유효한 프레전스 정보 데이터(presence attributes values linked with the synchronization key)를 비교하여 갱신된 영역(field)을 독출한다. 예를 들면, 친구리스트 중 3명이 온라인 상태에 있고, 3명을 제외한 친구들은 오프라인으로 되어 있는 상태에서 접속 해제되었다고 가정하면, 접속 해제 시 클라이언트 단말장치(10)와 서버

(110)는 각각 접속 해제시의 프레전스 정보 데이터를 저장하고 있고, 클라이언트 아이디와 세션 아이디 및 트랜잭션 아이디로 구성되는 동기 키값을 저장하고 있다. 이후 재접속할 시 클라이언트 단말장치(10)는 동기 키값을 포함시켜 프레전스 정보를 서버(110)로 요청하고, 이를 수신한 서버(110)는 접속 해제시에 저장되어 있는 프레전스 정보인 친구리스트 중 3명이 온라인 상태에 있고, 3명을 제외한 친구들은 오프라인으로 되어 있는 상태와 현재 친구들의 상태를 비교한다. 즉, 접속 해제시에는 온라인이었던 친구가 오프라인이 되었는지 아이디 및 주소 등의 상태정보가 바뀌었는지를 검사하여 바뀐 정보만을 클라이언트 단말장치(10)로 전송한다. 그러나 접속해제 시에 저장되어 있던 프레전스 정보 데이터와 재접속시 프레전스 정보 데이터 간에 업데이트(update) 된 정보가 없는 경우에는 응답 데이터 영역에 널(null) 상태로 데이터 전송하여 클라이언트 단말장치(10)로 업데이트한 프레전스 정보가 없음을 알린다. 동기화 과정이 완료되면 사용된 동기 키 값과 그에 대응되는 프레전스 데이터를 제거하여도 된다.

<41> 한편, 506단계와 512단계로부터 진행한 522단계에서는 클라이언트 단말장치(10)로부터 일반적인 프레전스 정보 동기화 요청이 있는지 검사하여 일반적인 프레전스 정보 동기화 요청이 있는 경우에는 524단계로 진행하여 현재 모든 프레전스 정보 데이터를 전송하고 종료한다.

<42> 상기한 바와 같이 본 발명은 메신저 서비스를 위해 클라이언트가 서버로 재접속 시 모든 프레전스 정보를 수신하는 것이 아니라 접속해제 시 저장된 프레전스 정보에서 업데이트 된 부분의 정보만을 수신하기 때문에 접속 시점에서 빠른 속도로 동기화할 수 있고, 적은 양의 데이터 송수신으로 인해 트래픽을 감소시킬 수 있다.

【발명의 효과】

- <43> 상술한 바와 같이 본 발명은 클라이언트가 서버로 재접속하는 경우 클라이언트와 서버가 동기화 할 필요가 없는 데이터들을 상호간에 교환하지 않아도 되므로 클라이언트, 서버 사이의 데이터 전송량이 적어지는 이점이 있다. 또한, 클라이언트가 서버에 접속 시 사용자의 친구 리스트에 포함된 친구들의 현재 상태 데이터를 빠른 시간안에 유효한 값으로 바꿀 수 있어 동기화 시간을 줄일 수 있는 이점이 있다.

【특허청구범위】**【청구항 1】**

메신저 서비스를 제공하는 시스템에 있어서,

상기 메신저 서비스를 위해 이전에 접속하였던 서버에 재접속하는 경우 업데이트 되어야 할 프레전스 정보 데이터를 상기 서버로 요청하여 상기 서버로부터 수신하는 업데이트 되어야 할 프레전스 정보 데이터에 의해 이전의 접속에 따라 저장되어 있는 프레전스 정보 데이터를 업데이트 하는 클라이언트 단말장치와,

상기 클라이언트 단말장치로부터 상기 업데이트 되어야 할 프레전스 정보 데이터 요구에 따라 현재 프레전스 정보 데이터 중 상기 이전의 접속 해제 이후부터 업데이트 된 프레전스 정보 데이터만을 상기 클라이언트 단말장치로 전송하는 서버로 구성됨을 특징으로 하는 시스템

【청구항 2】

제 1항에 있어서,

상기 서버에 연결되고, 상기 메신저 서비스를 위한 프레전스 정보들을 저장하는 프레전스 정보 데이터 베이스를 더 구비함을 특징으로 하는 시스템.

【청구항 3】

제 1항에 있어서, 상기 클라이언트 단말장치가,

현재 저장되어 있는 프레전스 정보 데이터에서 업데이트 할 프레전스 정보 데이터를 상기 서버로 요청하기 위해 접속해제 시 저장되어 있는 세션 아이디와 클라이언트 아이디와 트랜잭션 아이디로 구성되는 동기 키값을 생성하여 상기 생성된 동기 키값을 상기 서버로 전송함을 특징으로 하는 시스템.

【청구항 4】

제 3항에 있어서, 상기 세션 아이디는 상기 서버와 상기 클라이언트 단말장치의 이전 연결의 고유한 아이디이고, 상기 클라이언트 아이디는 상기 프레전스 정보 데이터를 저장하고 있는 클라이언트 단말장치의 고유한 아이디이고, 상기 트랜잭션 아이디는 접속 해지 전 상기 클라이언트 단말장치와 상기 서버간의 한번의 요청신호와 응답신호가 송수신된 마지막 트랜잭션의 아이디임을 특징으로 하는 시스템.

【청구항 5】

메신저 서비스를 제공하는 시스템에 있어서,

상기 메신저 서비스를 위해 이전에 접속하였던 서버에 재접속하는 경우 현재 저장되어 있는 세션 아이디와 클라이언트 아이디와 상기 서버와 상기 클라이언트 사이의 통신 중 정상적으로 종료된 트랜잭션 아이디로 구성되는 동기 키값을 생성하여 상기 생성된 동기 키값을 이용하여 업데이트 할 프레전스 정보 데이터를 상기 서버로 요청하고, 상기 서버로부터 수신하는 업데이트 되어야 할 프레전스 정보 데이터에 의해 이전의 접속에 따라 저장되어 있는 프레전스 정보 데이터를 업데이트 하는 클라이언트 단말장치와,

상기 클라이언트 단말장치로부터 상기 동기 키값이 수신되면 상기 동기 키값과 동일한 동기 키값이 있는 경우 상기 동기 키에 따라 저장되어 있는 프레전스 정보 데이터와 현재 프레전스 정보 데이터를 비교하여 업데이트 된 프레전스 정보 데이터를 독출하고, 상기 업데이트 된 프레전스 정보 데이터만을 상기 클라이언트 단말장치로 전송하는 서버와,

상기 서버에 연결되고, 상기 메신저 서비스를 위한 프레전스 정보들을 저장하는 프레전스 정보 데이터 베이스를 구비함을 특징으로 하는 시스템.

【청구항 6】

제 5항에 있어서, 상기 세션 아이디는 상기 서버와 상기 클라이언트 단말장치의 이전 연결의 고유한 아이디이고, 상기 클라이언트 아이디는 상기 프레전스 정보 데이터를 저장하고 있는 클라이언트 단말장치의 고유한 아이디이고, 상기 트랜잭션 아이디는 접속 해지 전 상기 클라이언트 단말장치와 상기 서버간의 한번의 요청신호와 응답신호가 송수신된 마지막 트랜잭션의 아이디임을 특징으로 하는 시스템.

【청구항 7】

제 6항에 있어서, 상기 서버는,

상기 수신한 동기 키와 동일한 동기 키가 상기 프레전스 정보 데이터 베이스에 저장되어 있으면, 현재의 프레전스 정보 데이터(presence attributes values at this moment)와 동기 키에 대응되어 저장된 접속 해제시 가장 마지막으로 유효한 프레전스 정보 데이터(presence attributes values linked with the synchronization key)를 비교하여 업데이트 된 영역

(field)을 독출하고, 상기 업데이트 된 영역을 프레전스 정보 요청에 대한 응답 데이터 영역에 포함시켜 전송함을 특징으로 하는 시스템.

【청구항 8】

제 7항에 있어서,

상기 업 데이트된 프레전스 정보 데이터가 없는 경우에는 상기 응답 데이터 영역에 널 (null) 상태로 전송함을 특징으로 하는 시스템.

【청구항 9】

클라이언트 단말장치와 상기 클라이언트 단말장치로 메신저 서비스를 제공하는 서버로 구성되는 시스템에서 상기 클라이언트 단말장치와 상기 클라이언트 단말장치간에 프레전스 정보 데이터 동기화를 위한 방법에 있어서,

상기 메신저 서비스를 위해 이전에 접속하였던 서버에 재접속하는 경우 업데이트 되어야 할 프레전스 정보 데이터를 상기 서버로 요청하는 과정과,

상기 서버는 상기 클라이언트 단말장치로부터 상기 업데이트 되어야 할 프레전스 정보 데이터 요구에 따라 현재 프레전스 정보 데이터 중 상기 이전의 접속 해제 이후부터 업데이트 된 프레전스 정보 데이터만을 상기 클라이언트 단말장치로 전송하는 과정과,

상기 클라이언트 단말장치가 상기 서버로부터 수신하는 업데이트 되어야 할 프레전스 정보 데이터에 의해 이전의 접속에 따라 저장되어 있는 프레전스 정보 데이터를 업데이트 하는

과정으로 이루어짐을 특징으로 하는 방법.

【청구항 10】

제 9항에 있어서, 상기 업데이트 할 프레젠테이션 정보 데이터를 상기 서버로 요청하는 과정은,

상기 클라이언트 단말장치가 접속해제 시 저장되어 있는 세션 아이디와 클라이언트 아이디와 트랜잭션 아이디로 구성되는 동기 키값을 생성하여 상기 생성된 동기 키값을 상기 서버로 전송하는 과정임을 특징으로 하는 방법.

【청구항 11】

제 9항에 있어서, 상기 세션 아이디는 상기 서버와 상기 클라이언트 단말장치의 이전 연결의 고유한 아이디이고, 상기 클라이언트 아이디는 상기 프레젠테이션 정보 데이터를 저장하고 있는 클라이언트 단말장치의 고유한 아이디이고, 상기 트랜잭션 아이디는 접속 해지 전 상기 클라이언트 단말장치와 상기 서버간의 한번의 요청신호와 응답신호가 송수신된 마지막 트랜잭션의 아이디임을 특징으로 하는 방법.

【청구항 12】

클라이언트 단말장치와 상기 클라이언트 단말장치로 메신저 서비스를 제공하는 서버로 구성되는 시스템에서 상기 클라이언트 단말장치와 상기 클라이언트 단말장치간에 데이터 동기화를 위한 방법에 있어서,

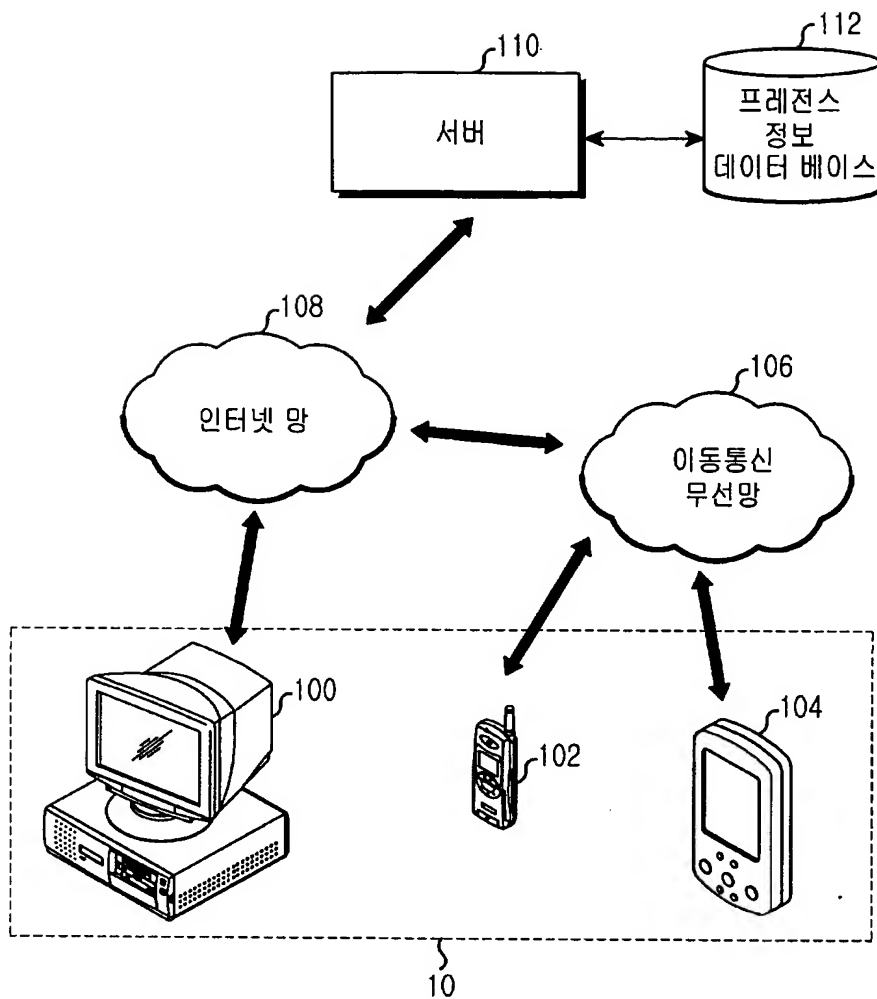
상기 클라이언트 단말장치가 상기 메신저 서비스를 위해 이전에 접속하였던 서버에 재 접속하는 경우 현재 저장되어 있는 세션 아이디와 클라이언트 아이디와 상기 서버와 상기 클라이언트 사이의 통신 중 정상적으로 종료된 트랜잭션 아이디로 구성되는 동기 키값을 생성하여 상기 생성된 동기 키값을 이용하여 업데이트 할 프레전스 정보 데이터를 상기 서버로 요청하는 과정과,

상기 서버가 상기 클라이언트 단말장치로부터 상기 동기 키값이 수신되면 상기 동기 키값과 동일한 동기 키값이 있는 경우 상기 동기 키에 따라 저장되어 있는 프레전스 정보 데이터와 현재 프레전스 정보 데이터를 비교하여 업데이트 된 프레전스 정보 데이터를 독출하고, 상기 업데이트 된 프레전스 정보 데이터만을 상기 클라이언트 단말장치로 전송하는 과정과,

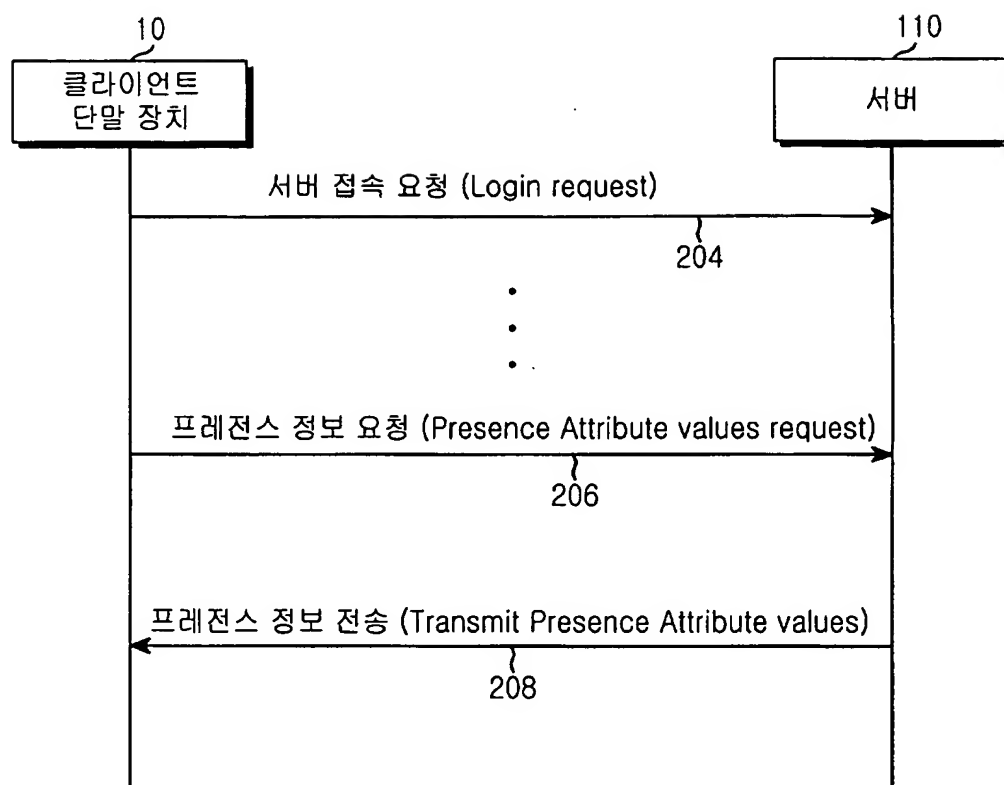
상기 클라이언트 단말장치가 상기 서버로부터 수신하는 업데이트 되어야 할 프레전스 정보 데이터에 의해 이전의 접속에 따라 저장되어 있는 프레전스 정보 데이터를 업데이트하여 상기 서버와 데이터 동기화시키는 과정으로 이루어짐을 특징으로 하는 방법.

【도면】

【도 1】

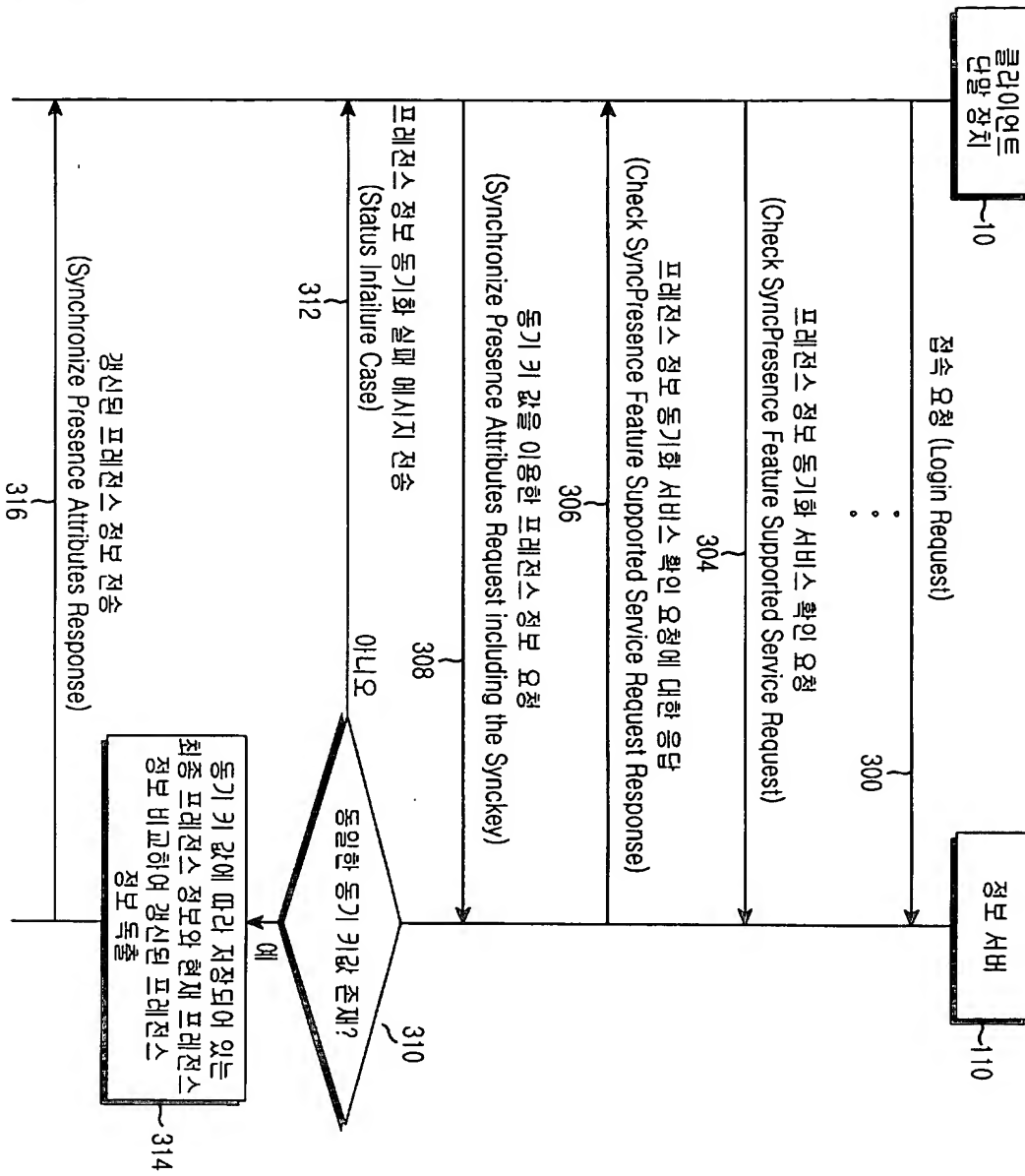


【도 2】

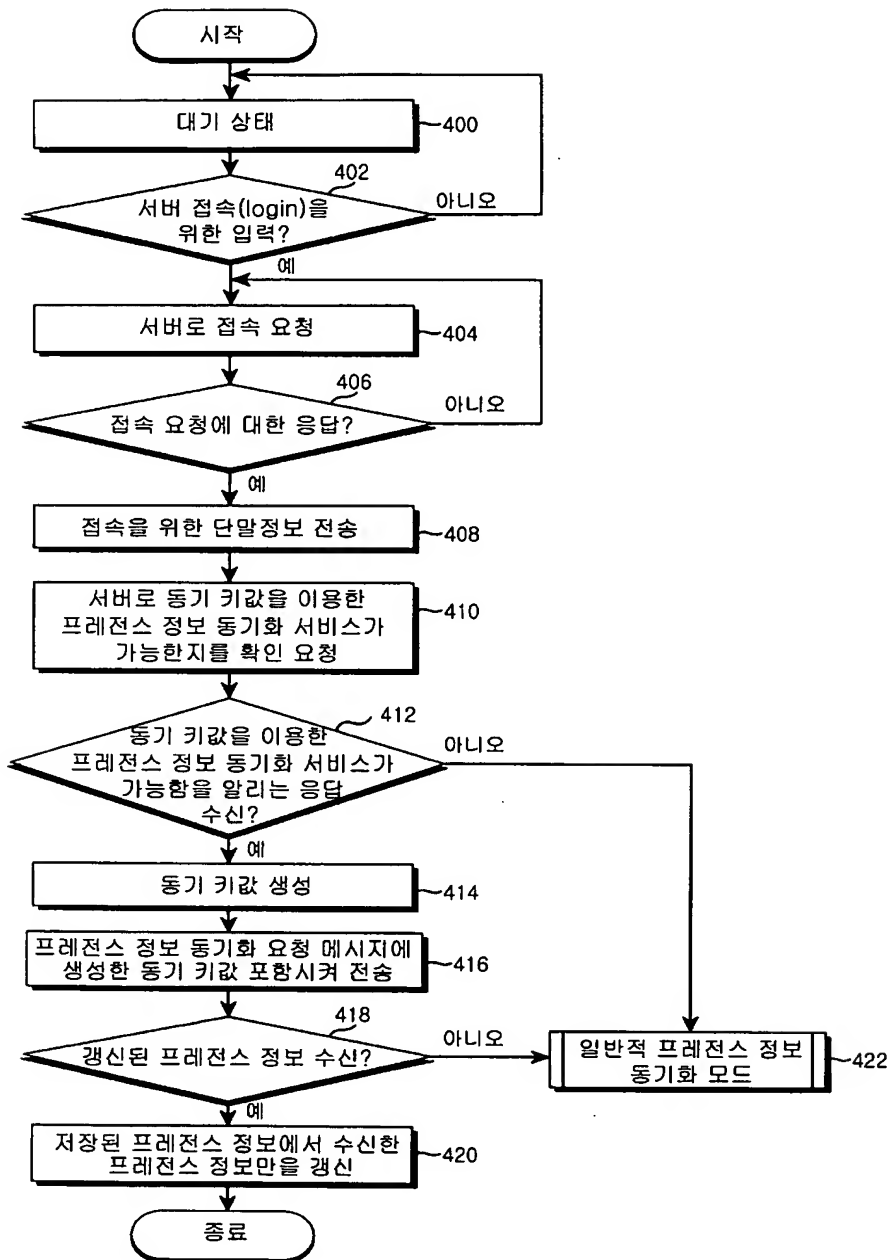




【도 3】



【도 4】



【도 5】

